Functional Inferences Over Heterogeneous Data

Functional Inferences Over Heterogeneous Data

Kwabena Nuamah, Alan Bundy, and Christopher Lucas

School of Informatics, University of Edinburgh
k.nuamah@ed.ac.uk,bundy@inf.ed.ac.uk,c.lucas@ed.ac.uk
http://www.inf.ed.ac.uk

Abstract. The increasing availability of knowledge bases (KBs) on the web has opened up the possibility of improved inference in automated query answering (QyA) systems. We have developed a rich inference framework (RIF) that responds to queries where no suitable answer is readily contained in any available data source, by applying functional inferences over heterogeneous data from the web. Our technique combines heuristics, logic and statistical methods to infer novel answers to queries. It also determines what facts are needed for inference, searches for them, and then integrates these diverse facts and their formalisms into a local query-specific inference tree. We explain the internal representation of RIF, the grammar and inference methods for expressing queries and the algorithm for inference. We also show how RIF estimates confidence in its answers, given the various forms of uncertainty faced by the framework.

1 Introduction

Inference enables an agent to create new knowledge from old. Our aim is to apply automatic inference to the semantic web, allowing users to extract new knowledge via queries, and dramatically increase the usefulness of semantic web data sources. RIF does not take natural language text as inputs. We use the acronym *QyA* for query answering, to distinguish it from question answering (QA) systems, which tend to focus on natural language processing (NLP) rather than inference of new facts. We focus on queries that require making predictions based on known facts about the past. We evaluate RIF in the domain of open governance, particularly, demography, education and agriculture. We use data from sources such as Wikidata [1], World Bank Data (WBD)(http://data.worldbank.org) and Geonames (http://www.geonames.org).

Our claim is that *the quality and range of answers generated by a query answering system is significantly improved when we automatically curate data and use rich forms of inference to infer novel knowledge from Semantic Web data and other semi-structured data from the web.* We use the term "rich" to emphasize the fact that the RIF relies on inference methods that go beyond first-order logic. We incorporate higher-order inference, where reasoning about functions expands the range of answers that can be sought. For instance, we can use regression to first construct functions then apply them to make predications. Answers to most queries in Table 2 can only be inferred by such prediction. Our view on

QyA complements NLP-driven approaches by inferring non-trivial answers from readily available facts in KBs by applying such rich forms of inference.

Information retrieval has been explored in different ways using techniques that include NLP and formal logic. Data sources from which answers are sought also range from logically represented facts, to natural language text on the Internet. The Semantic Web (SW) [2] offers practical approaches to representing shared knowledge across multiple domains on the Internet. Public agencies are responding to the initiative for open data by publishing their data using SW technologies. However, most query answering systems are still unable to effectively use these KBs to find answers that require inference beyond the retrieval of facts.

Systems such as AskMSR [4], START [5], Wolfram|Alpha(wolframalpha.com), PowerAqua [6], ANGIE [7], and OQA[8] are limited when the required facts are not stored in the KB. GORT [9], although it uses inference, is also heavily dependent on human input of missing facts and does not handle inference over functions. The systems surveyed in the QA tasks of the QALD (Question Answering over Linked Data) [10] challenge do not decompose queries beyond the NLP parse trees. For instance, in [11], the approach taken to answer questions with statistical linked data uses the NLP parse tree to generate the required SPARQL queries. Inference is, therefore, limited to the NLP parse since the process bottoms out at the SPARQL queries that are generated from it. Recent NLP techniques, such as dependency-based compositional semantics (DCS) [12], use statistical techniques that involve semantic parsing of questions to logical forms and evaluation of the logical forms with respect to a database of facts.

We use techniques from SW, logic and statistical inference to build the RIF.

2 The Rich Inference Framework

RIF uses a graph-based algorithm that recursively decomposes queries into sub-queries, eventually grounding out in either stored facts or previously cached answers. The decomposition at each level, as well as the means for combining sub-queries, is determined by features of the query or the sub-query's parent.

Facts retrieved from the external KBs used by the framework are primarily based on RDF [3] and are queried using the SPARQL query language or specific web APIs provided by the sources. This information needs to be curated to enrich it for the inference that is to follow. We augment the *subject(subj)*, *predicate(pred)*, *object(obj)* triple found in RDF KBs with *frames* that contain additional elements such as time, uncertainty, units of quantities, and other features as required. A frame is a list of *key:value* pair elements with keys that include (but are not limited to) *subj, pred, obj, time* and *confidence*. For example, the frame *[method:VALUE, subj:uk, pred:population, obj:63182000, time:2011, confidence:0.35]* represents the population of the UK in 2011 and the confidence RIF has in this fact.

2.1 Definitions

Definition 1. RIF Node: *A RIF node is a frame with elements whose values contain variables or ground terms.*

Variables indicate the elements of the frame to be looked up or inferred. Variables are prefixed with the $ or ? symbols. All variables in a RIF node are bound. Variables whose values are returned from a node are prefixed with the ? symbol. For instance, *[method:MAX, subj:uk, pred:population, obj:?y, time:2020]* shows that the object, y, is unknown and must be inferred and returned. An answer is found when all variables are instantiated to ground terms.

Definition 2. RIF Tree: *A tree of RIF nodes where each child node is derived from a decomposition of its parent node.*

RIF performs inference on a tree in two directions: (1) top-down: decomposing nodes using inference strategies, (2) bottom-up: using inference methods to propagate values from the leaf nodes back up the RIF tree to the root. Decomposition strategies label the arcs. Inference methods label the nodes.

Definition 3. Inference Method: *A higher-order function that aggregates values from a set of RIF nodes.* For instance, for a given node n_{parent}, its child nodes n^i_{child} and inference method, Σ_I, $n_{parent}.obj = \Sigma_I\{n^i_{child}.obj\}$, where $x.obj$ is the *object* element of the RIF node, x.

We use the notation $frame.key$ to extract the value of the specified element from the frame of a given RIF *node*.

In RIF, methods applied to RIF also return RIF nodes. An inference method, first extracts relevant values from its child RIF nodes to use as inputs and then applies the function associated with the method. The inference method then substitutes the inferred value into the respective RIF node elements and returns the complete RIF node. Methods used in RIF are listed in Table 1.

Definition 4. Query: *A query is a composition of inference methods and contains both functional and propositional logics for describing entities and relations.* Functional is used at meta-level for inference methods and propositional for the object logic. We use a context-free grammar in Extended Backus-Naur Form (BNF) that, together with the type signatures of inference methods, defines well-formed queries. RIF queries take the form:

```
func_expr :: METHOD_NAME((var|<var,var>),(logic_expr|func_expr)[,(logic_expr|func_expr)])
```

where *var* are variables, $func\_expr$ are functional expressions and *logic_expr* represents propositional expressions. Examples of are shown in Table 2. The convention is that methods are all caps and propositional constants can begin with either a lower or upper case.

Predicates are not pre-defined prior to their use in RIF. The framework finds matching predicates in the KBs from which the corresponding subjects (or objects) are retrieved. The matching process uses string functions to split words in a predicate, language resources such as WordNet [13] to find synonyms, and edit distance measures to find matches to predicates in a KB.

Definition 5. Inference Strategy: An inference strategy (decomposition) is a transformation on a RIF node from which child nodes are derived.

For a given RIF node, n_{parent} and *strategy*, S, the decomposition, Δ, is the mapping: $\Delta_S(n_{parent}) \mapsto \{n^i_{child} | i > 0\}$

Strategies used in RIF include: the *temporal* strategy, to decompose nodes by date/time features; *geospatial* strategy, to decompose nodes by location; and the *lookup* strategy, to create child nodes with synonyms of the elements of the parent, to increases the chances of finding facts in KB.

Table 1. Inference Methods

Method	Description
VALUE	Default method. Returns value of node
SUM	Add values of nodes of numeric type
AVG	Mean value of child nodes
MEDIAN	Median value of child nodes
MAX	Maximum value of child nodes
MIN	Minimum value of child nodes
COMP	Obtain a set by list comprehension
GT	'Greater Than' function to compare two nodes
LT	'Less Than' function to compare two nodes
EQ	Check if the value of two nodes are equal
REGRESS	Regression function from child nodes.
LOOKUP	Find facts from knolwedge bases.

Table 2. Types of queries and examples

Type 1: Facts Retrieval
Q1.What was the urban population of UK in 2010? *VALUE(?y,urban_population(Uk,?y,2010))*
Type 2: Aggregation
Q2.Which country had the lowest female unemployment in South America in 2011? *MIN($y,COMP(⟨?x,$y⟩,female_unemployment(?x,$y,2011): Country(?x) & location(?x,South_America)))*
Type 3: Nested Queries
Q3.Was the rural population of the country with the largest arable land in Africa greater than the urban population of the country with the smallest arable land in Africa in 2003? *GT(?b,VALUE(?b,rural_population(MAX($d, COMP(⟨?c,$d⟩,arable_land(?c,$d,2003): Country(?c) & location(?c,Africa))),?b,2003)), VALUE(?b,urban_population(MIN($h, COMP(⟨?g,$h⟩,arable_land(?g,$h,2003):Country(?g) & location(?g,Africa))),?b,2003)))*
Type 4: Prediction
Q4.What was the GDP in 2010 of the country predicted to have the largest total population in Europe in 2018? *VALUE(?y,gdp(MAX($b,COMP(⟨?a,$b⟩, population(?a,$b,2018):Country(?a) & location(?a,Europe))),?y,2010)))*

2.2 Implementation

RIF explores strategies and executes the necessary inference methods to infer a novel answer from the available facts. Fig. 1 illustrates RIF with different inference strategies that recursively define new nodes and inference methods that infer answers that are propagated up the RIF tree. It is shown as an *AND-OR* graph where the OR branches show strategy options and the AND branches show node decompositions.

RIF begins with the root node as a goal and searches for facts in the KB that match variables in the node. If no fact is found, RIF selects the appropriate strategies to create child nodes. Strategies are selected based in on features in queries such as names of location or date/time. For each child node with an answer that is resolved, the answer is propagated back to the parent which in turn aggregates its child nodes and infers a new fact based on its inference method. This process continues until an answer is propagated to the root goal. To prevent unnecessary decompositions, we set a depth limit on the graph. If the inference tree depth bound is reached and no relevant facts are found in the KB at the leaves, the framework yields no answer. For nested queries (e.g. Q3 in Table 2), the parent node spawns child nodes to solve the sub queries.

Due to the time overheads in calling web services, we store a local copy of KBs such as WordNet, ConceptNet [14] and Geonames, which are used frequently by RIF and are rarely changed by their authors. However, data resources such as WBD and the Scottish Government Data (http://statistics.gov.scot/), that are

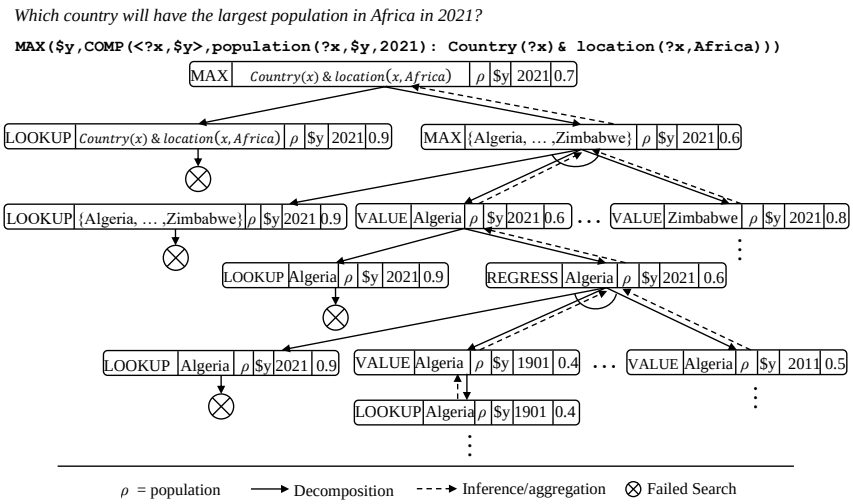

Fig. 1. RIF Example: Initial decompositions (showing AND-OR search tree). The frame of each node is represented compactly as *[method,subj,pred,obj,time,confidence]*.

frequently updated, are queried directly using APIs provided by their publishers. In these cases, we cache facts retrieved and inferred as well as inferred functions.

We implemented RIF in Java. We also set up a local RDF triplestore using Apache Jena (https://jena.apache.org) to host frequently used KBs such as Geonames and used a MongoDB (www.mongodb.org) instance for caching.

2.3 Uncertainty in RIF

RIF has two main sources of uncertainty: (1) credibility of KBs and noise in the data, and (2) errors introduced by the inference methods. We have initially focused on the first form of uncertainty and limit it to real-valued facts. We will tackle the latter form in future work. The confidence of a node captures the uncertainty in an estimate, normalized by its magnitude, i.e., the *coefficient of variation (CoV)* or σ/μ, where σ is the estimated standard deviation and μ is the posterior mean. This currently applies to positive real-valued facts. We assume that each retrieved real-valued fact is an observation of the true value with additive Gaussian noise, where the noise variance depends on the assumed credibility of the source. RIF estimates the confidence in an answer by combining and propagating the confidence values of child nodes to their parent recursively in closed form. We use a normal approximation in the estimation and propagation of confidence.

3 Evaluation

We tested our hypothesis by evaluating RIF with a variety of queries (github.com/knuamah/rif). Concretely, we focused on queries that are of interest to public

Table 3. Evaluation results by queries types, showing the percentage of queries answered successfully.

Query Types	1	2	3	4	Overall
RIF(%)	90	80	80	70	80

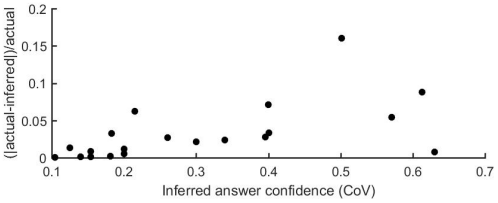

Fig. 2. CoV and estimation error plot.

institutions that publish open data on the web. We based the test queries on real-valued facts available in Wikidata and the WBD.

Existing test sets for evaluating query answering systems focus on aspects of the inference process that differ from our objective with RIF. We were interested in queries that, not only find relevant facts, but also infer non-trivial answers by combining them. We therefore compiled questions that are usually asked about demographics and other country development indicators. Our evaluation consisted of forty queries spanning the four main query types shown in Table 2. Results for the four query types are shown in Table 3. We also used cross-validation to evaluate the confidence scores estimated by RIF. We compared the absolute difference between the inferred value and the true 'held-out' fact to the confidence score (CoV). Results are shown in Fig.3. We obtained good results in both tests.

RIF's use of geospatial, temporal and commonsense facts as well as higher order functions allowed it to tackle the range of test queries with 80% overall success. RIF's main limitation was its word matching mechanism, where it failed to find the appropriate matches from KBs in some cases. This was due to its lack of NLP to handle the useful NL descriptions contained in facts. Hence, when the same fact was provided in multiple units, RIF easily mixed them up.

Finally, our evaluation of the confidence scores estimated by RIF also showed a good correlation between the confidence score (CoV) and the error between the true fact and what was inferred.

4 Conclusion

Our Rich Inference approach to QyA enables us to increase the range of queries that can be answered by a QyA system to include prediction and interpolation. The framework also estimates its confidence in the answers inferred given the underlying data and methods used for inference. Finally, the inference trees generated give full access to how answers were inferred. This, we believe, makes our approach practically useful for users who wish to verify answers.

In future work, we plan to extend the algorithm to incorporate the confidence scores in the selection and prioritization of inference strategies, as well as capture non-positive-real-valued data, such as booleans and discrete values, in confidence estimations. We will also consider uncertainty arising from approximations made by inference methods.

References

1. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM. vol. 57, pp. 78–85. Springer (2014)
2. Berners-Lee, T., Hendler, J.: The Semantic Web. Scientific American. vol.284, (2001)
3. Beckett, D., McBride, B.: RDF/XML syntax specification (revised). W3C recommendation. vol. 10 (2004)
4. Banko, M., Brill, E., Dumais, S., Lin, J.: AskMSR: Question Answering Using the Worldwide Web. Proceedings of 2002 AAAI Spring Symposium on Mining Answers, pp. 1–2. (2002)
5. Katz, B.: Annotating the World Wide Web Using Natural Language. Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet (RIAO '97) (1997)
6. Lopez, V., Fernández, M., Motta, E., Stieler, N.: PowerAqua: Supporting users in querying and exploring the Semantic Web. Semantic Web. vol. 3, pp. 249–265. (2012)
7. Preda, N., Kasneci, G.: Active knowledge: dynamically enriching RDF knowledge bases by web services. SIGMOD, Indianapolis. (2010)
8. Fader, A., Zettlemoyer, L., Etzioni, O.: Open question answering over curated and extracted knowledge bases. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14. pp. 1156–1165. ACM Press (2014)
9. Bundy, A., Sasnauskas, G., Chan, M.: Solving Guesstimation Problems Using the Semantic Web : Four Lessons from an Application. Semantic Web. pp. 1–20. (2013)
10. Unger, C., et al.: Question answering over linked data (QALD-5). Working Notes for CLEF 2015 Conference, (2015)
11. Höffner, K., Lehmann, J.: Towards question answering on statistical linked data. Proceedings of the 10th International Conference on Semantic Systems. ACM (2014)
12. Liang, P., Jordan, M., Klein, D.: Learning dependency-based compositional semantics. Computational Linguistics. vol. 39, pp. 389–446. MIT Press (2013)
13. Miller, G.A.: WordNet: a lexical database for English. Communications of the ACM. vol. 38, pp. 39–41. ACM (1995)
14. Liu, H., Singh, P.: ConceptNet - a practical commonsense reasoning tool-kit. BT technology journal. vol. 22, pp. 211–226. Springer (2004)